PCT WORLD INTELLECTUAL PROPERTY ORGANIZATION
International Bureau

INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classification 6 : G06F 17/30

A1

(11) International Publication Number: **WO 99/66429**

(43) International Publication Date: 23 December 1999 (23.12.99)

(21) International Application Number: PCT/US99/13829

(22) International Filing Date: 18 June 1999 (18.06.99)

(30) Priority Data:
09/099,954 19 June 1998 (19.06.98) US

(71) Applicant: OZ INTERACTIVE, INC. [US/US]; 525 Brannan Street, San Francisco, CA 94107 (US).

(72) Inventor: ÞORKELSSON, Haraldur, Haukur; Viðimelur 50, IS–107 Reykjavik (IS).

(74) Agents: SCHNELLER, John, W. et al.; Venable, P.O. Box 34385, Washington, DC 20043–9998 (US).

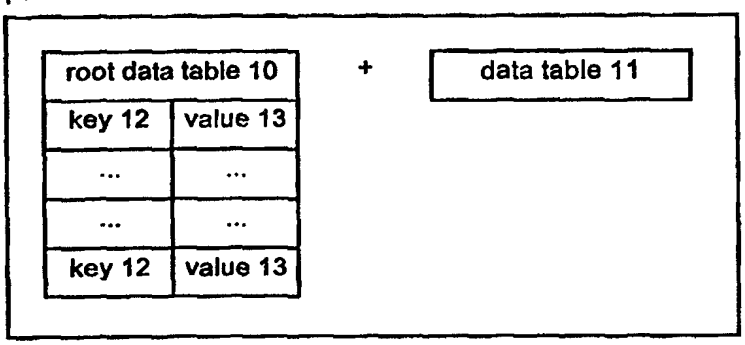
(81) Designated States: AU, CA, JP, European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE).

Published
With international search report.
Before the expiration of the time limit for amending the claims and to be republished in the event of the receipt of amendments.

(54) Title: NETWORK FOR A VIRTUAL ENVIRONMENT

(57) Abstract

A network connects clients and servers for maintaining a hierarchical distributed database. The hierarchical distributed database is implemented using partitionable relational databases, each of which is assigned to a server on the network. Using the hierarchical distributed database, a virtual environment can be created for simultaneous real–time access by millions of clients.



FOR THE PURPOSES OF INFORMATION ONLY

Codes used to identify States party to the PCT on the front pages of pamphlets publishing international applications under the PCT.

AL	Albania	**ES**	Spain	**LS**	Lesotho	**SI**	Slovenia
AM	Armenia	**FI**	Finland	**LT**	Lithuania	**SK**	Slovakia
AT	Austria	**FR**	France	**LU**	Luxembourg	**SN**	Senegal
AU	Australia	**GA**	Gabon	**LV**	Latvia	**SZ**	Swaziland
AZ	Azerbaijan	**GB**	United Kingdom	**MC**	Monaco	**TD**	Chad
BA	Bosnia and Herzegovina	**GE**	Georgia	**MD**	Republic of Moldova	**TG**	Togo
BB	Barbados	**GH**	Ghana	**MG**	Madagascar	**TJ**	Tajikistan
BE	Belgium	**GN**	Guinea	**MK**	The former Yugoslav Republic of Macedonia	**TM**	Turkmenistan
BF	Burkina Faso	**GR**	Greece			**TR**	Turkey
BG	Bulgaria	**HU**	Hungary	**ML**	Mali	**TT**	Trinidad and Tobago
BJ	Benin	**IE**	Ireland	**MN**	Mongolia	**UA**	Ukraine
BR	Brazil	**IL**	Israel	**MR**	Mauritania	**UG**	Uganda
BY	Belarus	**IS**	Iceland	**MW**	Malawi	**US**	United States of America
CA	Canada	**IT**	Italy	**MX**	Mexico	**UZ**	Uzbekistan
CF	Central African Republic	**JP**	Japan	**NE**	Niger	**VN**	Viet Nam
CG	Congo	**KE**	Kenya	**NL**	Netherlands	**YU**	Yugoslavia
CH	Switzerland	**KG**	Kyrgyzstan	**NO**	Norway	**ZW**	Zimbabwe
CI	Côte d'Ivoire	**KP**	Democratic People's Republic of Korea	**NZ**	New Zealand		
CM	Cameroon			**PL**	Poland		
CN	China	**KR**	Republic of Korea	**PT**	Portugal		
CU	Cuba	**KZ**	Kazakstan	**RO**	Romania		
CZ	Czech Republic	**LC**	Saint Lucia	**RU**	Russian Federation		
DE	Germany	**LI**	Liechtenstein	**SD**	Sudan		
DK	Denmark	**LK**	Sri Lanka	**SE**	Sweden		
EE	Estonia	**LR**	Liberia	**SG**	Singapore		

NETWORK FOR A VIRTUAL ENVIRONMENT

FIELD OF THE INVENTION

The invention relates generally to networks of computers and, in particular, to distributing and maintaining a database over a network which is able to be accessed by millions of clients simultaneously.

BACKGROUND OF THE INVENTION

Conventionally, network applications are designed to communicate using either client-server interaction or peer-to-peer interaction. Although client-server interaction in general provides a possibility for high reliability and accessibility, the client-server interaction can not be scaled to serve large numbers of users on the order of more than a million. Problems with scalability of the client-server interaction can conventionally be partially fixed by applying replication of the server's database. Replication, however, can lead to data inconsistency among the replicated services.

Data inconsistencies on the replicated server are conventionally addressed by using a method to prevent data inconsistency or using a method to resolve the data inconsistency. A method to prevent data inconsistency typically involves locking, which takes time to execute, permits only one client to access the locked data, and can cause dead locks. To implement locking, the client determines that to fulfill the user's command, the client needs to update a common data state of the sever. To ensure no one else can change the common data state of the server while the client is accessing it, the client requests a lock on the data from the server. If the data is not already locked by someone else, the server locks the data and sends out a message to the client stating the client and only the client can now change the data. If the data is already locked, the client must wait for the lock to be released. Once the server acknowledges that the data is locked for exclusive use by the client, the client sends the modification for the data to the server. The server sends the changed data back to the client. Finally, the client sends a lock release message to the server.

Although some of the steps can be simplified, the problem with locking remains the same. Locking is time consuming and limits the number of clients that can modify the data at any one time to one in a real-time environment. This is unacceptable because it provides a finite number of maximum clients, which is determined by: (the

average time to modify data)*(the maximum number of modifications need per timeslot)/(the minimum time slot size for real-time performance).

Locking can also cause deadlocks. For example, client A locks data A and then determines that it needs data B. Client B, however, at the same time locks data B and then determines that it needs data A to continue. Both clients A and B end up waiting forever, unless some deadlocking prevention is added. Deadlocking prevention, however, is time consuming and affects real-time performance. Hence, the conventional methods to prevent data inconsistency can not be used in a real-time environment because of the disruption that the deadlocks cause to the client-server interaction.

As another method to address problems with data inconsistency, resolving data inconsistency typically uses transaction time stamps. Such a method, however, results in lengthy rollbacks of the client-server interactions causing a break in the real-time environment and possibly overloading the system if too many rollbacks occur.

Further, the rollbacks cause unacceptable behavior in a virtual environment as experienced by the affected users. For example, a user walks through a door in a virtual environment, and the user's client determines that the door is open, but later finds out that the door was locked by another user just before the user walked through the door. The client and server then need to drag the user back out of the room just entered because the user's being in the room does not consist with what other users presume to be accurate for the virtual environment. Hence, in the user's view of the virtual environment, events will be abstract when the user is bounced back through the door.

In addition to client-server interaction, networked applications can be designed to communicate using peer-to-peer interaction. Peer-to-peer interaction, however, creates problems with controlling ownership and maintaining continuous data states when the users of the networked application log off. Further, peer-to-peer interaction also has problems with being scalable similar to that of the client-server interaction as discussed above. Moreover, because of the relatively limited bandwidth between the clients, and because a server is not filtering the data transfer between the clients, much data is transferred, and the transfer takes a significant amount of time to accomplish, causing a deterioration in the real-time environment.

SUMMARY OF THE INVENTION

In discussing the invention, the following defined terms are used throughout.

"Distributed database" refers to a database maintained and distributed over a plurality of computers connected via a network.

"Hierarchical distributed database" refers to a distributed database having a hierarchical arrangement to its distribution.

"Network" refers to a system of connected computers. Non-limiting examples of a network include: a local area network ("LAN"); a wide area network; a broad band network, such as the network connecting automatic teller machines ("ATMs"); and the Internet.

"Client" refers to a computer application running on one or more computers and accessing a network.

"Server" refers to a computer for controlling interactions on a network.

"Virtual environment" refers to a three-dimensional graphics computer application, a virtual reality computer application, or a virtual computer environment.

"User" refers to a human interacting with a computer.

"Computer" refers to a computer, a general purpose computer, an interactive television, a hybrid combination of a computer and an interactive television, and any apparatus comprising a processing unit, memory, the capability to receive input, and the capability to generate output. One or more computers connected with a network can maintain a virtual environment accessible by one or more clients.

It is an object of the invention to provide a distributed database for access by many users in real-time, where the distributed database is both scalable and hierarchical.

Another object of the invention is to provide a network and a server for maintaining a distributed database.

A further object of the invention is to maintain a virtual environment over a network using a distributed database.

Yet another object of the invention is to provide a distributed database accessible by many users via a network.

An additional object of the invention is to provide a system for maintaining a virtual environment using a network of servers that not only distributes the load of the virtual environment among the servers but also keeps the cohesion of the virtual environment as high as possible.

Still another object of the present invention is to provide a network that is scalable to accommodate an expanding hierarchical distributed database.

The above objects and advantages of the invention are achieved by an apparatus and a method. The apparatus of the invention includes a network for maintaining a hierarchical distributed database comprising: a plurality of servers connected to the network, each server comprising a partitionable relational database; and a plurality of clients connected to the network, each client accessing the hierarchical distributed database; wherein the hierarchical distributed database comprises the partitionable relational databases of the plurality of servers, the partitionable relational databases being arranged hierarchically; wherein each client accesses the hierarchical distributed database by connecting to one of the servers; and wherein each client can access the hierarchical distributed database without waiting for another client to finish accessing the hierarchical distributed database.

The apparatus of the invention also includes a server connected to a network, the network comprising: a hierarchical distributed database, the hierarchical distributed database being distributed over the network and arranged hierarchically; and the server comprising a partitionable relational database comprising a data table, the partitionable relational database being connected hierarchically to the hierarchical distributed database, the data table comprising at least one key-value pair comprising a key data structure and a value data structure, each value data structure comprising a data structure or a link for linking to the hierarchical distributed database hierarchically arranged below the partitionable relational database.

The method of the invention includes a method for maintaining over a network a hierarchical distributed database having a plurality of partitionable relational databases, comprising the steps of: providing a plurality of servers and a plurality of clients; arranging the partitionable relational databases of hierarchical distributed database in a hierarchy; assigning each of the partitionable relational databases of hierarchical distributed database to the plurality of servers; and permitting simultaneous access to the hierarchical distributed database by the plurality of clients.

Moreover, the above objects and advantages of the invention are illustrative, and not exhaustive, of those which can be achieved by the invention. Thus, these and other objects and advantages of the invention will be apparent from the description herein or can be learned from practicing the invention, both as embodied here and as modified in view of any variations which may be apparent to those skilled in the art.

BRIEF DESCRIPTION OF THE DRAWINGS

The invention is described in greater detail below by way of reference to the accompanying drawings, where similar reference characters refer to similar reference parts throughout the drawings:

Figure 1 illustrates the components of a server for the invention.

Figure 2 illustrates a partitionable relational database for the server illustrated in Figure 1.

Figure 3 illustrates an example of a partitionable relational database using the invention.

Figure 4 illustrates the functionality of the server executable kernel of Figure 1.

Figure 5 illustrates a hierarchical distributed database with two levels of servers, namely top level servers and second level servers.

Figure 6 illustrates server connections for the hierarchical distributed database of Figure 5.

Figure 7 illustrates alternative server connections for the hierarchical distributed database of Figure 5.

Figure 8 illustrates server connections for the hierarchical distributed database of the invention.

DETAILED DESCRIPTION OF THE INVENTION

In the invention, a hierarchical distributed database is maintained over a network comprising clients and servers. The hierarchical distributed database is scalable and is able to support a virtual environment simultaneously accessible in real-time by a plurality of clients. Dependent only on having enough space in the virtual environment, a few clients, over a million clients, and up to a limitless number of clients can access the hierarchical distributed database of the invention.

With the invention, a client in the network interacts only with a set of related data. The relation can be on any scale. For example, one server can maintain a database of rooms in a building, and additional servers can maintain things in the rooms in the building, and so on. This configuration both distributes the load of the database and keeps the total network load of a single transaction limited to only a fraction of the network. In such a network, the clients can view the part of the database that is relevant to them and only that part, thus making the system scalable. The network of the

invention is expandable to serve an ever increasing hierarchical distributed database, such as a virtual environment, by increasing the size of the network.

In Figure 1, a server 1 for the invention is illustrated and comprises several components. The multicast data transmission protocol 2 of the server 1 provides for transmitting data over the network to the hierarchical distributed database and receiving data over the network from the hierarchical distributed database. The multicast data transmission protocol minimizes network traffic of transactions and data streams.

With the multicast data transmission protocol 2, if a client requests to be updated on changes to a particular data, the client will be registered as a receiver of the particular data. In a network, data changes rapidly, and a client must continually request the data by, for example, reading or polling the data. However, with multicasting, a client can simply register for updates on the changes to the data. Using multicast channels, a state change in the virtual environment is distributed in a single event, like a selective broadcast, over the network to the client who has requested the information. For example, if two clients on server 2 have requested information on the change in data on server 1, server 1 only needs to send the requested data to server 2, and server 2 then forwards the data to the two clients on server 2. This feature of the invention advantageously decreases traffic on the network when there are many clients.

The partitionable relational database 3 is a component of the hierarchical distributed database assigned to the server 1. Preferably, the partitionable relational database 3 of the server 1 is stored in the memory of the computer comprising the server 1.

In Figure 2, the partitionable relational database 3 of server 1 of Figure 1 is illustrated. The partitionable relational database 3 comprises at least one table. The at least one table comprises a root data table 10 and zero or more additional data tables 11. The root data table 10 and the data tables 11 each comprise an ordered list of one or more key-valued pairs. The key 12 of the key-value pair functions as an address for the key-value pair.

The value 13 of the key-value pair can be any type of data structure. Examples of the value 13 include: a web page for an Internet site; a file written in the hypertext markup language ("HTML"), the virtual reality markup language ("VRML"), or VRML2; a virtual environment description; a user description; an avatar description; an

avatar position; a binary data file; text; and a numerical value. "Avatar" refers to the representation of a user within a virtual environment.

The value 13 of the key-value pair can also be a link for linking, or a pointer for pointing, to a table located on the same server or located anywhere in the network.

Figure 3 illustrates an example of a partitionable relational database using the invention. The various connections between the tables of the database are illustrated. For example, the table "Info stored on server" has a link to the table "Users and Info."

The authentication system 4 of the server 1 in Figure 1 authenticates data transmissions received by the server 1 over the network. The authentication system 4 is connection based and, as such, the client is authenticated when first connecting to the network. The client then has an authentication key it uses throughout the session on the network. The authentication key is used, for example, when connecting to another server in the network. The authentication key maps the client to its client information on the server, which can then confirm who the client is and use this client information for access control.

The distributed access control system 5 of the server 1 provides for accessing the hierarchical distributed database on the network by securing ownership and access to data contained within the hierarchical distributed database. Every access to any data, such as tables and key-value pairs, can be restricted to, for example, read, modify, delete, and administrator. For all data, access can be granted or denied by default, and an access table lists those clients with default access. An access table is a list of clients or groups of clients who have been granted or denied a specific type of access to the respective data. During a session on the network, a client requesting to perform actions on a data, such as read or modify, is validated for access according to the default of the access table. Once validation is completed, information about the validation is added to the authentication key of the client.

The distributed atomic transaction interface 6 of the server 1 provides for interfacing with the network and enables the real-time interaction with the hierarchical distributed database. An atomic transaction is a transaction that is performed in a single step. For example, for the atomic transaction of a client command to modify a data entity, the server responds positively or negatively after performing the action in a single step, without performing other actions at the same time. This feature of the

invention prevents locking, which occurs in conventional networks. In addition, the distributed atomic transaction interface 6 includes support for security.

The server executable kernel 7 of the server 1 is a program running on the computer of the server 1 for overseeing the operations of the server 1. The kernel 7 is responsible for overseeing the multicasting, routing, and connections of the server 1 to clients and to other servers and for maintaining the partitionable relational database 3 of the server 1.

Figure 4 illustrates the functionality of the kernel 7 within the server 1.

Figure 5 illustrates the hierarchical distributed database of the invention. The hierarchical distributed database of Figure 5 has two layers, a top layer for top level servers and a second layer for second level servers. The top layer is shown with one top level server, and the second layer is show with two second level servers. The top level server in Figure 5 has two tables, a "Users and info" table and a "regions handled by network" table. The second level server on the right side of Figure 5 has three tables, a "Users on server" table, a "Region 1" table, and a "Users in region" table. The second level server on the left side of Figure 5 has five tables, a "Users on server" table, a "Region 1" table, a "Users in region" table for region 1, a "Region 2" table, and a "Users in region" table for region 2.

As illustrated in Figure 5, the hierarchical distributed database of the invention is arranged in a hierarchical fashion. Further, the hierarchical distributed database of the invention is distributed and maintained on a plurality of servers.

Figure 6 illustrates an abstracted view of the relationship between the servers and the connections in Figure 5. The two second level servers, Server 2 and Server 3, are connected to the top level server, Server 1. The entry point for a client to access the hierarchical distributed database is through the top level server, Server 1, and then down the database to either Server 2 or Server 3, depending on the services and information needed.

With the invention, any number of layers and servers can be connected in a variety of ways using the invention. For example, Figure 7 illustrates an alternative to the relationship shown in Figure 6. Server 3 is connected to Server 2, and not to Server 1, as in Figure 6. In this arrangement, there are three levels of servers.

Further, Figure 8 illustrates an alternative to the relationship shown in Figures 6 and 7. Compared to Figures 6 and 7, the servers in Figure 8 are no longer arranged in a

hierarchical fashion. However, like Figures 6 and 7, the distributed database in Figure 8 is still hierarchical.

When the database is being distributed over the servers, care should be taken with respect to the relationships in the virtual environment and the clients' data access needs in the virtual environment. For example, if a client can enter regions 1, 2, or 3, the server connections illustrated in Figure 6 are useful. As another example, if a client can only enter regions 2 and 3 through region 1, the server connections illustrated in Figure 7 are useful. As an additional example, if regions 2 and 3 contain information on clients in region 1, as in the case of a chat room, the server connections illustrated in Figure 7 are useful.

To access the hierarchical distributed database, several of the servers maintaining the tables of the hierarchical distributed database are designated as top level servers, or root servers. Top level servers are the entry points by which clients can access the hierarchical distributed database. After entering the hierarchical database via a top level server, a client can logically "walk down" the structure of the database to access information relevant to the client's processing needs. Preferably, the top level servers are designated as the servers located close to the tables needed by clients.

Although the database maintained by the servers is arranged hierarchically, the internal representation of the database in the client is not restricted to being hierarchical. The internal representation maintained by the client of the hierarchical distributed database maintained by the servers can be any internal representation based on the client's needs or on the construction of the client's databases.

With the hierarchical distributed database of the invention, a server in the network only needs to be connected to the servers in the network that have tables to which the tables in the server have links. Hence, with the invention, a limitless network can be created by logically arranging the tables of the hierarchical distributed database such that no single table is larger than one server can handle and by connecting the clients to the top level servers. In this manner, the hierarchical distributed database is scalable and can support a few clients, over a million clients, and up to a limitless number of clients.

With the invention, a real-time virtual environment can be created for simultaneous real-time access by a few clients, by over a million clients, by millions of clients, and up to a limitless number of clients. Each user of the virtual environment has

a client for accessing the hierarchical distributed database of the invention. Further, with the invention, real-time interaction between users in the virtual environment occurs.

In a virtual environment using the invention, a single server can be accessed by multiple clients, and a single server can maintain a state within the virtual environment. For example, on a single server using the invention, many users represented by avatars can interact within a room of a virtual environment.

The invention has been described in detail with respect to preferred embodiments, and it will now be apparent from the foregoing to those skilled in the art that changes and modifications may be made without departing from the invention in its broader aspects, and the invention, therefore, as defined in the following claims is intended to cover all such changes and modifications as fall within the true spirit of the invention.

CLAIMS

What is claimed is:

1. A network for maintaining a hierarchical distributed database comprising:

a plurality of servers connected to the network, each server comprising a partitionable relational database; and

a plurality of clients connected to the network, each client accessing the hierarchical distributed database;

wherein the hierarchical distributed database comprises the partitionable relational databases of the plurality of servers, the partitionable relational databases being arranged hierarchically;

wherein each client accesses the hierarchical distributed database by connecting to one of the servers; and

wherein each client can access the hierarchical distributed database without waiting for another client to finish accessing the hierarchical distributed database.

2. A network according to claim 1, wherein a subset of the plurality of servers are designated as top level servers; and

wherein each client accesses the hierarchical distributed database by connecting to one of the top level servers.

3. A network according to claim 1, wherein the partitionable relational database of each server comprises a data table;

wherein each data table comprises a key-value pair; and

wherein each key-value pair comprises a key data structure and a value data structure.

4. A network according to claim 4, wherein the value data structure of one of the key-value pairs comprises a data structure.

5. A network according to claim 3, wherein the value data structure of one of the key-value pairs comprises a link for linking to the hierarchical distributed database arranged hierarchically below the partitionable relational database of the value data structure.

6. A network according to claim 1, wherein the hierarchical distributed database is a real-time virtual environment.

7. A network according to claim 6, wherein the clients are represented by avatars in the real-time virtual environment.

8. A network according to claim 1, wherein each server further comprises means for carrying out a multicast data transmission protocol for transmitting over the network modifications to the hierarchical distributed database.

9. A network according to claim 1, wherein each server further comprises:

an authentication system for authenticating data transmissions received by the server;

a distributed access control system for accessing the hierarchical distributed database; and

a distributed atomic transaction interface for interfacing with the network.

10. A server connected to a network, the network comprising:

a hierarchical distributed database, the hierarchical distributed database being distributed over the network and arranged hierarchically; and

the server comprising:

a partitionable relational database comprising a data table, the partitionable relational database being connected hierarchically to the hierarchical distributed database, the root data table comprising at least one key-value pair comprising a key data structure and a value data structure, each value data structure comprising a data structure or a link for linking to the hierarchical distributed database hierarchically arranged below the partitionable relational database.

11. A server according to claim 8, the server further comprising:

means for carrying out a multicast data transmission protocol for transmitting data over the network modifications to the hierarchical distributed database;

an authentication system for authenticating data transmissions received by the server;

a distributed access control system for accessing the hierarchical distributed database; and

a distributed atomic transaction interface for interfacing with the network.

12. A server according to claim 8, wherein the distributed database is a real-time virtual environment.

13. A method for maintaining over a network a hierarchical distributed database having a plurality of partitionable relational databases, comprising the steps of:

providing a plurality of servers and a plurality of clients;

arranging the partitionable relational databases of hierarchical distributed database in a hierarchy;

assigning each of the partitionable relational databases of hierarchical distributed database to the plurality of servers; and

permitting simultaneous access to the hierarchical distributed database by the plurality of clients.

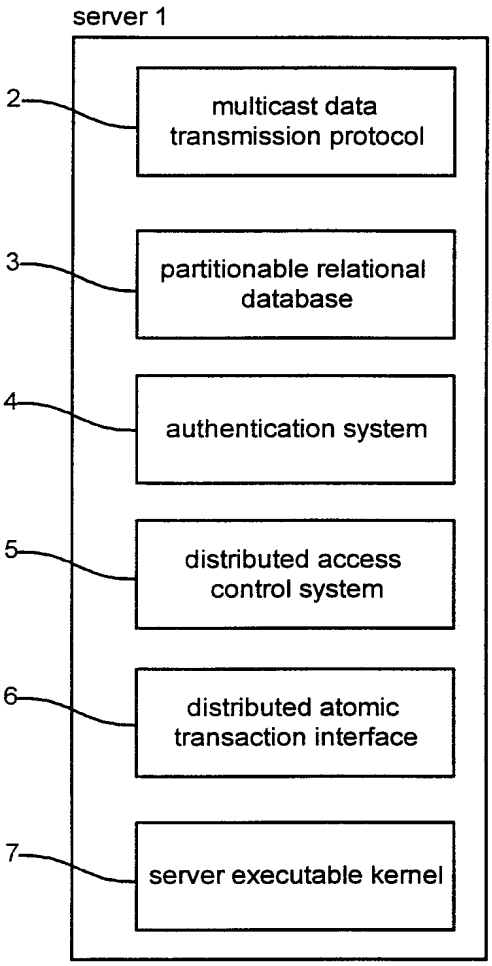

server 1
2
multicast data transmission protocol
3
partitionable relational database
4
authentication system
5
distributed access control system
6
distributed atomic transaction interface
7
server executable kernel

Figure 1

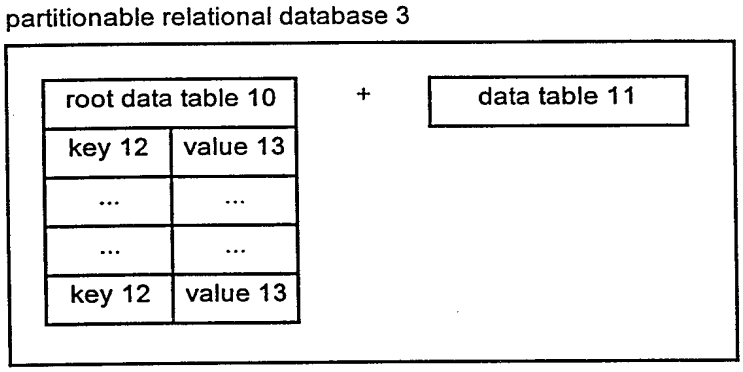

partitionable relational database 3
root data table 10
key 12
value 13
...
...
...
...
key 12
value 13
+
data table 11

Figure 2

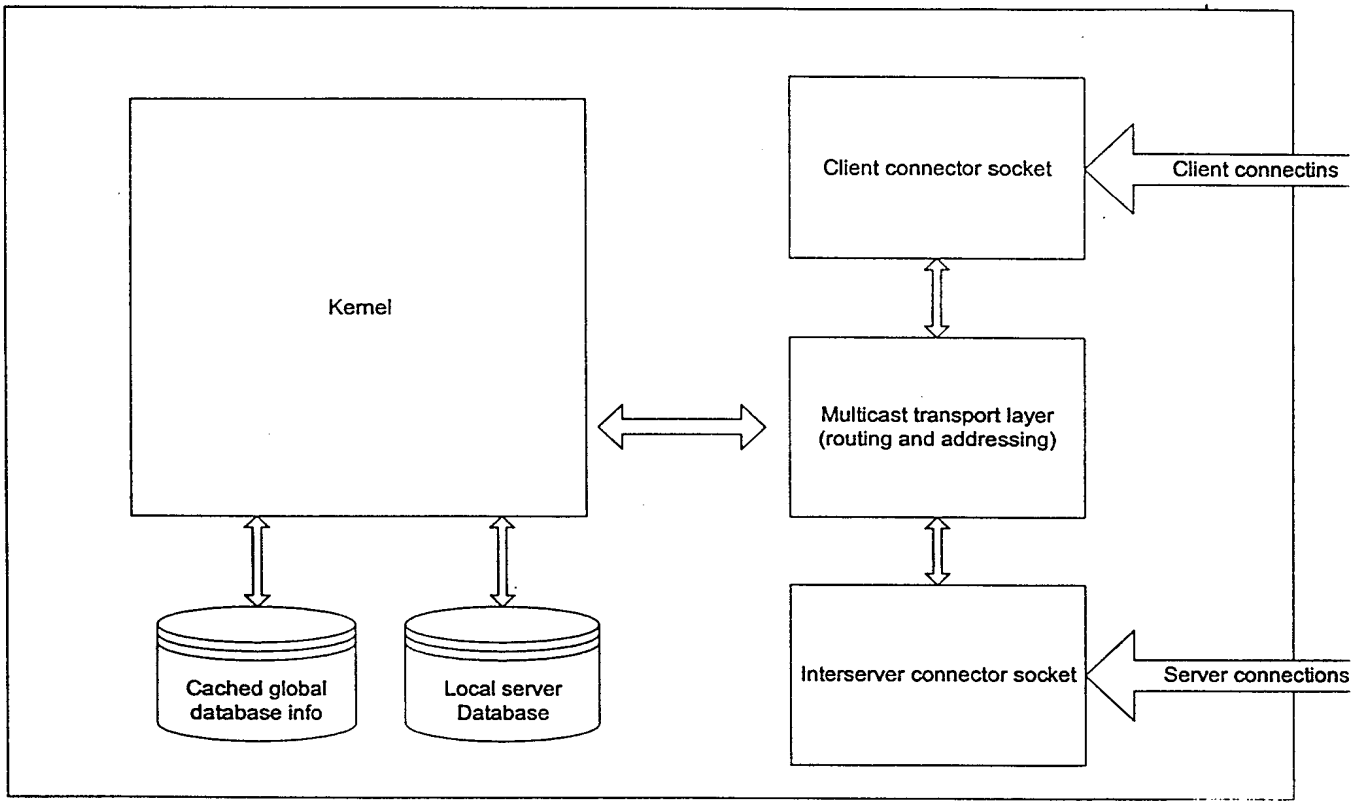

Kernel
Client connector socket
Client connectins
Multicast transport layer
(routing and addressing)
Interserver connector socket
Server connections
Cached global
database info
Local server
Database

Figure 3

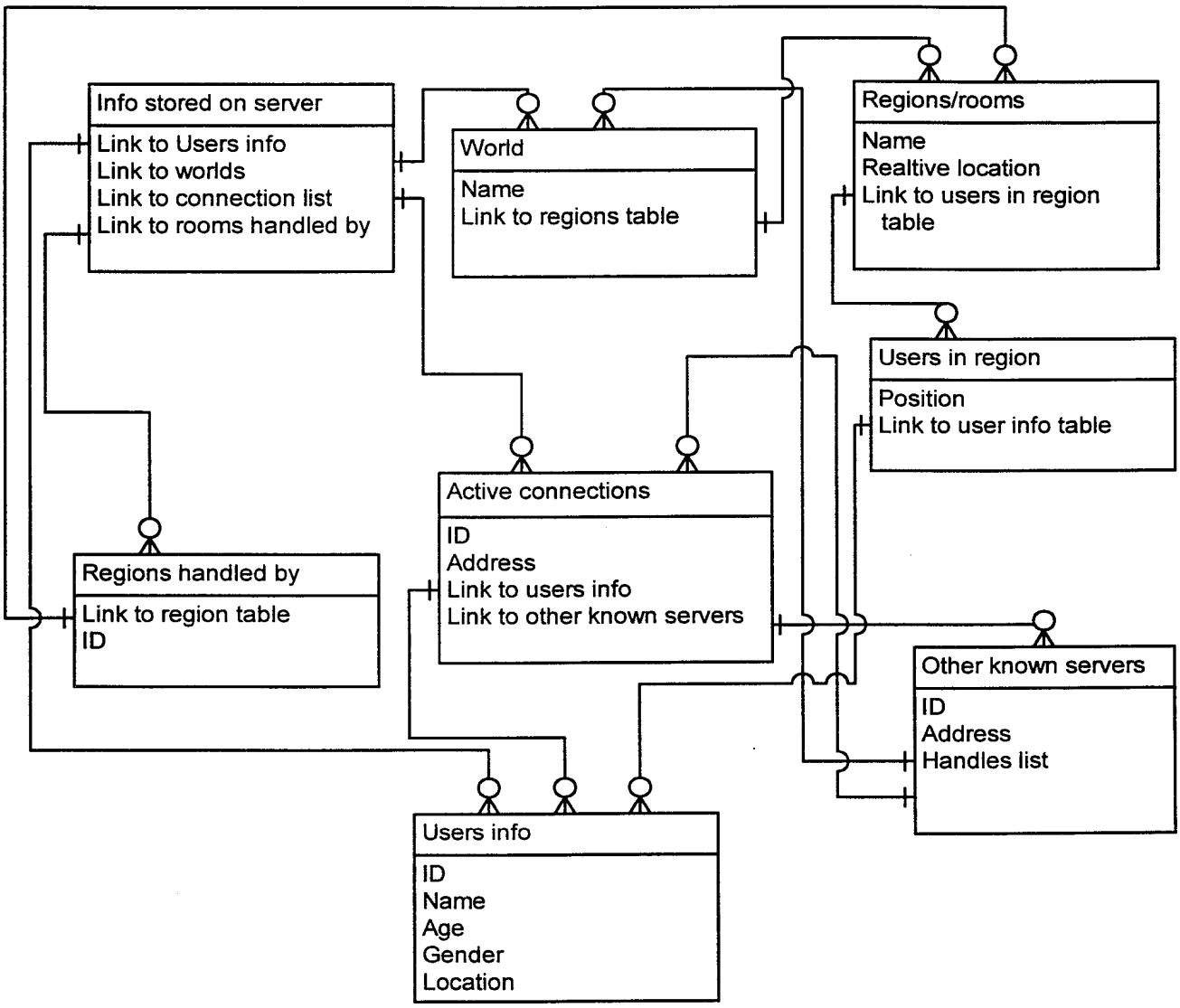

Info stored on server
Link to Users info
Link to worlds
Link to connection list
Link to rooms handled by
World
Name
Link to regions table
Regions/rooms
Name
Realtive location
Link to users in region table
Users in region
Position
Link to user info table
Active connections
ID
Address
Link to users info
Link to other known servers
Regions handled by
Link to region table
ID
Other known servers
ID
Address
Handles list
Users info
ID
Name
Age
Gender
Location

Figure 4

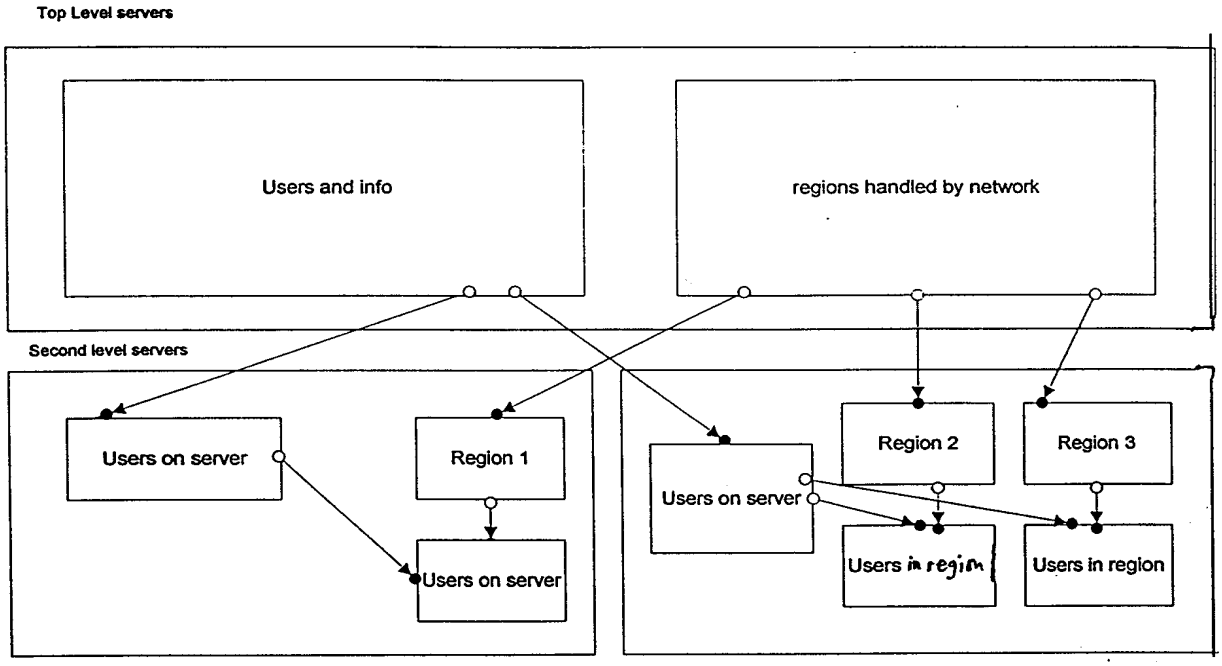

Top Level servers
Users and info
regions handled by network
Second level servers
Users on server
Region 1
Users on server
Users on server
Region 2
Region 3
Users in region
Users in region

Figure 5

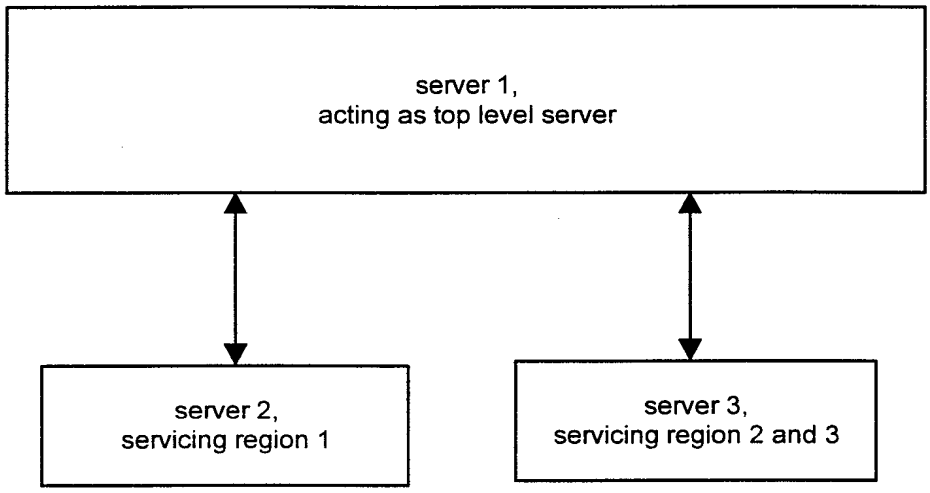


Figure 6

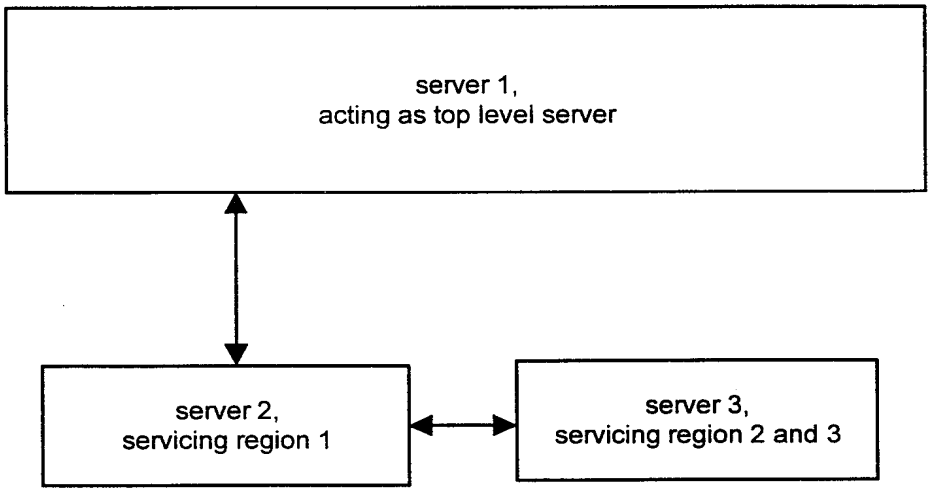


Figure 7

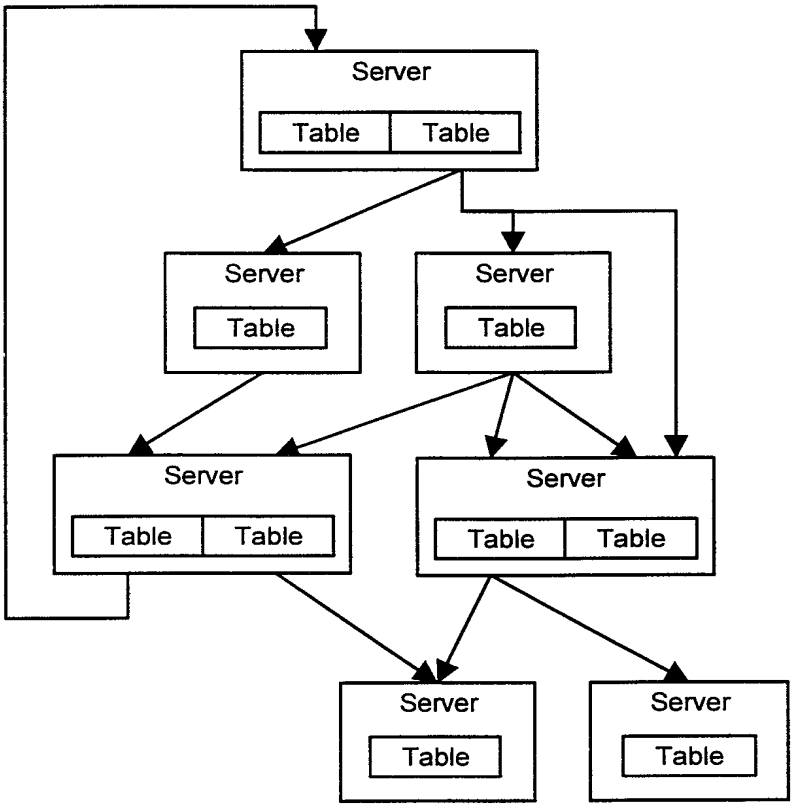

Server
Table
Table
Server
Table
Server
Table
Server
Table
Table
Server
Table
Table
Server
Table
Server
Table

Figure 8

INTERNATIONAL SEARCH REPORT

Int ational Application No
PC /US 99/13829

A. CLASSIFICATION OF SUBJECT MATTER
IPC 6 G06F17/30

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
IPC 6 G06F

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practical, search terms used)

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category °	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	WO 98 20430 A (CURRENT NETWORK TECHNOLOGIES C ;SUTTER HERBERT P (CA)) 14 May 1998 (1998-05-14) abstract; claim 1 page 3, line 12 -page 3, line 25 page 14, line 19 -page 14, line 30	1,10,13

☐ Further documents are listed in the continuation of box C.

☒ Patent family members are listed in annex.

° Special categories of cited documents :

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier document but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art.

"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
14 October 1999	21/10/1999

Name and mailing address of the ISA	Authorized officer
European Patent Office, P.B. 5818 Patentlaan 2 NL – 2280 HV Rijswijk Tel. (+31–70) 340–2040, Tx. 31 651 epo nl, Fax: (+31–70) 340–3016	Katerbau, R

1

INTERNATIONAL SEARCH REPORT

Information on patent family members

Int ational Application No
PC /US 99/13829

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
WO 9820430 A	14-05-1998	US 5924094 A	13-07-1999
		AU 4858497 A	29-05-1998
		EP 0934568 A	11-08-1999